

# Continuous Graph Partitioning for Camera Network Surveillance

D. Borra<sup>a</sup>, F. Pasqualetti<sup>b</sup>, F. Bullo<sup>b</sup>

<sup>a</sup>*Dipartimento di Scienze Matematiche, Politecnico di Torino, Italy*

<sup>b</sup>*Center for Control, Dynamical Systems, and Computation, University of California, Santa Barbara, USA*

---

## Abstract

In this work we design surveillance trajectories for a network of autonomous cameras to detect intruders in an environment. Intruders, which appear at arbitrary times and locations, are classified as *static* or *dynamic*. While static intruders remain stationary, dynamic intruders are aware of the cameras configuration and move to avoid detection, if possible. As performance criteria we consider the *worst-case detection time* of static and dynamic intruders. We model the environment and the camera network by means of a robotic roadmap. We show that optimal cameras trajectories against static intruders are obtained by solving a continuous graph partitioning problem. We design centralized and distributed algorithms to solve this continuous graph partitioning problem. Our centralized solution relies on tools from convex optimization. For the distributed case, we consider three distinct cameras communication models and propose a corresponding algorithm for each of the models. Regarding dynamic intruders, we identify necessary and sufficient conditions on the cameras locations to detect dynamic intruders in finite time. Additionally, we construct constant-factor optimal trajectories for the case of ring and tree roadmaps.

*Key words:* Graph partitioning, Convex constrained optimization, Camera network, Remote and distributed control.

---

## 1 Introduction

Remote surveillance of human activities for civil and military applications is receiving considerable attention from the research community. Public areas such as banks, art galleries, private houses, prisons, department stores, and parking lots, are now equipped with camera networks to detect important activities, [18, 22]. From a technological perspective, one of the main challenges consists of developing efficient algorithms for the cameras to autonomously and distributively complete tracking, surveillance, and recognition tasks.

In this work we focus on the problem of detecting intruders by means of a network of autonomous cameras. In particular, we consider Pan-Tilt-Zoom (PTZ) cameras installed at important locations. We assume the cameras to move their *field of view* (f.o.v.) to cooperatively surveil the whole environment. We develop algorithms for the cameras to self-organize and to detect *intruders* in the environment, that appear at arbitrary locations and times. We consider *static* intruders, which remain stationary, and *dynamic* intruders, which move to avoid detection, if possible. As performance criteria we consider the *worst-case detection time*, that is the longest time needed for the cameras to detect intruders.

**Related work.** Works related to our camera surveillance problem can be found in the mobile robotics and computer science literatures. In mobile robotics, the patrolling problem consists of scheduling the motion of a team of autonomous agents in order to detect intruders or important events, e.g., see [1, 4, 17, 20]. It should be observed

---

\* This work was supported by NSF grants IIS-0904501 and CNS-1035917, and by ARO grant W911NF-11-1-0092.

*Email addresses:* domenica.borra@polito.it (D. Borra), fabiopas@engineering.ucsb.edu (F. Pasqualetti), bullo@engineering.ucsb.edu (F. Bullo).

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE <b>23 JUL 2012</b>		2. REPORT TYPE		3. DATES COVERED <b>00-00-2012 to 00-00-2012</b>	
4. TITLE AND SUBTITLE <b>Continuous Graph Partitioning for Camera Network Surveillance</b>				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>University of California at Santa Barbara, Center for Control, Dynamical Systems and Computation, Santa Barbara, CA, 93106</b>				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT <b>Approved for public release; distribution unlimited</b>					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT <b>In this work we design surveillance trajectories for a network of autonomous cameras to detect intruders in an environment. Intruders, which appear at arbitrary times and locations, are classified as static or dynamic. While static intruders remain stationary, dynamic intruders are aware of the camera configuration and move to avoid detection, if possible. As performance criteria we consider the worst-case detection time of static and dynamic intruders. We model the environment and the camera network by means of a robotic roadmap. We show that optimal camera trajectories against static intruders are obtained by solving a continuous graph partitioning problem. We design centralized and distributed algorithms to solve this continuous graph partitioning problem. Our centralized solution relies on tools from convex optimization. For the distributed case we consider three distinct camera communication models and propose a corresponding algorithm for each of the models. Regarding dynamic intruders, we identify necessary and sufficient conditions on the camera locations to detect dynamic intruders in finite time. Additionally, we construct constant-factor optimal trajectories for the case of ring and tree roadmaps.</b>					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT <b>Same as Report (SAR)</b>	18. NUMBER OF PAGES <b>16</b>	19a. NAME OF RESPONSIBLE PERSON
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>			

that the patrolling problem and the problem considered in this paper significantly differ. Indeed, cameras are fixed at predetermined locations, and their f.o.v.s must lie within the cameras visibility constraints. On the other hand, robots are usually allowed to travel the whole environment, and are usually not subject to visibility constraints. Consequently, algorithms developed for teams of robots are, in general, not applicable in our setup. Similarly, algorithms for graph-clearing and graph-search do not extend to our scenario [13, 14, 19].

In the context of camera networks, the perimeter patrolling problem has recently been studied in [4, 8, 21]. In these works, distributed algorithms are proposed for the cameras to partition a one-dimensional environment, and to synchronize along a trajectory with minimum worst-case detection time of intruders. We improve the results along these directions by, for instance, developing cameras trajectories and partitioning methods for general environment topologies.

In this work we present algorithms for graph partitioning. It is worth noting that our graph partitioning problem differs from classical setups, e.g., see [2, 3, 11, 12]. Indeed, in these works the graph partitioning problem is usually combinatorial, and it consists of partitioning the vertices or the edges as to optimize a certain performance function. Instead, we formulate continuous graph partitioning problems, in which the graph is a physical entity, and the partition is obtained by splitting the edges. As it will be clear in the sequel, our results on graph partitioning are general and applicable to different problems. For instance, if each edge of the graph represents a task to be accomplished by the processors at its endpoints, then our algorithms can be used for dynamic load balancing for multiprocessor networks [9, 16].

A preliminary version of this work appeared in [6]. Extensions with this work include (i) the design of trajectories against static intruders for cyclic roadmaps, (ii) the design of trajectories against dynamic intruders, and (iii) a characterization of necessary and sufficient conditions for finite detection time of dynamic intruders.

**Paper contributions.** The main contributions of this work are as follows. First, we propose the *continuous graph partitioning problem*, in which a partition of a weighted graph is obtained by splitting the graph edges, and the cost of a partition equals the longest length of its parts (Section 2). We show that the continuous graph partitioning problem is convex and non-differentiable, and we characterize its solutions. Then, we derive an equivalent convex and differentiable partitioning problem, which is amenable to distributed implementation.

Second, we define the camera surveillance problem for the detection of static and dynamic intruders (Section 3). We model the environment and the camera network by means of a robotic roadmap, and we formalize the worst-case detection time of static and dynamic intruders.

Third, we exhaustively discuss the case of static intruders (Section 4). We show that, for tree and ring roadmaps, cameras trajectories with minimum worst-case detection time can be designed by solving a continuous graph partitioning problem. For general cyclic roadmaps, our trajectories based on continuous partitions are proved to be optimal up to a factor 2. However, we conjecture that optimality is achieved also in this case.

Fourth, for the case of dynamic intruders, we derive a necessary and sufficient condition on the cameras locations for the existence of a trajectory with finite detection time (Section 5). We focus on ring and tree roadmaps. In particular, for the case of ring roadmaps we design a trajectory with detection time within a factor  $3/2$  of optimal. Instead, for tree roadmaps, the performance of our trajectory is within a factor 2 of optimal.

Fifth and finally, we consider three different communication models, and we propose distributed algorithms for the cameras for continuous graph partitioning in all these scenarios (Section 6). In particular: our first algorithm assumes a synchronous mode of operation of the cameras; our second algorithm assumes an asymmetric broadcast communication model and extends the class of block-coordinate descent algorithms to the constrained case; and our third algorithm only requires gossip communication. We prove convergence of all these algorithms, and we analyze their performance in a simulation study.

## 2 Continuous Partitions of Weighted Graphs

In this section we introduce the problem of continuous graph partitioning. A solution to this problem will be used to design optimal cameras trajectories.

Let  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  be an undirected weighted graph, where  $\mathcal{V}$  and  $\mathcal{E}$  denote the vertex and edge sets, respectively. Let  $\ell_{ij} \in \mathbb{R}_{>0}$  be the weight associated with the edge  $\{v_i, v_j\} \in \mathcal{E}$ . For a subset of vertices  $\mathcal{V}_c \subseteq \mathcal{V}$ , define the  $i$ -th set of neighbors as  $\mathcal{N}_i = \mathcal{N}_i^{\text{in}} \cup \mathcal{N}_i^{\text{out}}$ , where

$$\begin{aligned}\mathcal{N}_i^{\text{in}} &= \{v_j \in \mathcal{V}_c : \{v_i, v_j\} \in \mathcal{E}\}, \\ \mathcal{N}_i^{\text{out}} &= \{v_j \in \mathcal{V} \setminus \mathcal{V}_c : \{v_i, v_j\} \in \mathcal{E}\}.\end{aligned}\tag{1}$$

A *continuous partition* of the weighted graph  $\mathcal{G}$  is a set  $\mathcal{P} = \{\mathcal{P}_1, \dots, \mathcal{P}_n\}$ , where (see Fig. 1)

$$\mathcal{P}_i = \bigcup_{v_j \in \mathcal{N}_i} [v_i, v_j],\tag{2}$$

and  $v_{ij} \in [v_i, v_j]$  is defined by some  $\alpha_{ij} \in [0, 1]$  as<sup>1</sup>

$$v_{ij} = \begin{cases} v_i + \alpha_{ij}(v_j - v_i), & \text{if } j \in \mathcal{N}_i^{\text{in}}, i < j, \\ v_i + (1 - \alpha_{ji})(v_j - v_i), & \text{if } j \in \mathcal{N}_i^{\text{in}}, i > j, \\ v_j, & \text{if } j \in \mathcal{N}_i^{\text{out}}. \end{cases}\tag{3}$$

The dimension of a continuous partition equals  $\mathcal{L}(\mathcal{P}) = \max\{L_1, \dots, L_n\}$ , where  $L_i$  is the sum of the lengths of the segments in  $\mathcal{P}_i, i = 1, \dots, n$ . In other words

$$L_i = \sum_{v_j \in \mathcal{N}_i^{\text{in}}, i < j} \alpha_{ij} \ell_{ij} + \sum_{v_j \in \mathcal{N}_i^{\text{in}}, i > j} (1 - \alpha_{ji}) \ell_{ij} + \sum_{v_j \in \mathcal{N}_i^{\text{out}}} \ell_{ij}.\tag{4}$$

Let  $L$  and  $\alpha$  be the vectors of  $L_i$  and  $\alpha_{ij}$ , respectively. Notice that a continuous partition is entirely specified by a parameters vector  $\alpha$ .

Let  $\mathcal{G}(\mathcal{V}_c) = (\mathcal{V}_c, \mathcal{E}_c)$  be the subgraph of  $\mathcal{G}$  induced by the vertices  $\mathcal{V}_c$ , where  $\mathcal{E}_c = (\mathcal{V}_c \times \mathcal{V}_c) \cap \mathcal{E}$ . Define the weighted *incidence matrix*  $A \in \mathbb{R}^{|\mathcal{V}_c| \times |\mathcal{E}_c|}$  as

$$A_{i,e} = \begin{cases} \ell_{ij}, & \text{if } e = \{v_i, v_j\} \in \mathcal{E}_c, i < j, \\ -\ell_{ij}, & \text{if } e = \{v_i, v_j\} \in \mathcal{E}_c, i > j, \\ 0, & \text{otherwise,} \end{cases}\tag{5}$$

and the weighted *incidence vector*  $\mathbf{b} \in \mathbb{R}^{|\mathcal{V}_c|}$  as

$$\mathbf{b}_i = \sum_{v_j \in \mathcal{N}_i^{\text{in}}, i > j} \ell_{ij} + \sum_{v_j \in \mathcal{N}_i^{\text{out}}} \ell_{ij}.\tag{6}$$

Notice that  $L = A\alpha + \mathbf{b}$ , and that for every  $\alpha \in \mathbb{R}^{|\mathcal{E}_c|}$  it holds

$$\|A\alpha + \mathbf{b}\|_1 = \sum_{\{v_i, v_j\} \in \mathcal{E}} \ell_{ij}.$$

Let  $\mathbf{0}$  and  $\mathbf{1}$  be the vectors of all zeros and ones, respectively. We address the following minimization problem.

**Problem 1 (Continuous min-max partition)** *Given a weighted graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  and a subset of vertices  $\mathcal{V}_c \subseteq \mathcal{V}$ , let  $A$  and  $\mathbf{b}$  be as in (5) and (6), respectively. Determine a continuous partition  $\alpha_\infty^*$  satisfying*

$$\|A\alpha_\infty^* + \mathbf{b}\|_\infty = \min_{\alpha \leq \alpha_\infty^* \leq \bar{\alpha}} \|A\alpha + \mathbf{b}\|_\infty,\tag{7}$$

<sup>1</sup> Given any two points  $x_i, x_j \in \mathbb{R}^m$  for some  $m \in \mathbb{N}$ , we let  $[x_i, x_j] = \{tx_i + (1-t)x_j : t \in [0, 1]\}$ .

for some constraints vectors  $\underline{\alpha}, \bar{\alpha}$  with  $\mathbf{0} \leq \underline{\alpha} \leq \bar{\alpha} \leq \mathbf{1}$ .

Notice that (7) is a convex minimization problem, for which efficient centralized solvers exist [7]. On the other hand, since (7) is not differentiable, distributed solvers may be difficult to implement. We next derive an equivalent differentiable minimization problem, which is instead amenable to distributed implementation.

**Problem 2 (Continuous min partition)** *Given a weighted graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  and a subset of vertices  $\mathcal{V}_c \subseteq \mathcal{V}$ , let  $A$  and  $\mathbf{b}$  be as in (5) and (6), respectively. Determine a continuous partition  $\alpha_2^*$  satisfying*

$$\|A\alpha_2^* + \mathbf{b}\|_2 = \min_{\underline{\alpha} \leq \alpha \leq \bar{\alpha}} \|A\alpha + \mathbf{b}\|_2. \quad (8)$$

for some constraints vectors  $\underline{\alpha}, \bar{\alpha}$  with  $\mathbf{0} \leq \underline{\alpha} \leq \bar{\alpha} \leq \mathbf{1}$ .

**Remark 3 (Uniqueness of partitions)** *Since the minimization problem (8) is strictly convex, the continuous min partitioning problem admits a unique minimum value, and the set of minimizers is a singleton if and only if the matrix  $A$  has a trivial null space. It can be shown that  $A$  has a trivial null space if and only if the induced graph  $\mathcal{G}(\mathcal{V}_c)$  is a tree. In particular, if the graph  $\mathcal{G}(\mathcal{V}_c)$  is connected, then the dimension of the null space of  $A$  equals  $|\mathcal{E}_c| - |\mathcal{V}_c| + 1$ .*

**Remark 4 (Unconstrained partitions)** *Consider the (unconstrained) partitioning problems (7) and (8) with  $\underline{\alpha}, \bar{\alpha} \in \mathbb{R}^{|\mathcal{E}_c|}$ . It can be verified that  $\alpha^*$  is a minimizer of Problem (7) if and only if it is a minimizer of Problem (8). Moreover, every minimizer can be written as  $\alpha^* = A^\dagger(\mathbf{v} - \mathbf{b}) + \mathbf{w}$ , where  $\mathbf{v} = \left(\sum_{\{v_i, v_j\} \in \mathcal{E}} \ell_{ij} / n\right) \mathbf{1}$ ,  $A^\dagger$  is the pseudoinverse of  $A$ ,  $\mathbf{w}$  satisfies  $A\mathbf{w} = \mathbf{0}$  and  $\mathbf{0} \leq \alpha^* \leq \mathbf{1}$ .*

We next show a relation between min-max partitions and min partitions.

**Theorem 5 (Min-max and min partitions)** *Let  $\alpha_2^*$  be a min partition solution to Problem 2. Then,  $\alpha_2^*$  is also a solution to Problem 1, that is,*

$$\|A\alpha_2^* + \mathbf{b}\|_\infty = \min_{\underline{\alpha} \leq \alpha \leq \bar{\alpha}} \|A\alpha + \mathbf{b}\|_\infty.$$

In order to prove Theorem 5, we introduce the following definitions and results. Given a partition  $\mathcal{P} = \{\mathcal{P}_1, \dots, \mathcal{P}_n\}$  defined by  $\alpha$ , the *maximal graph* associated with  $\alpha$  is  $\mathcal{G}^{\max} = (\mathcal{V}_c^{\max}, \mathcal{E}_c^{\max})$ , where  $\mathcal{V}_c^{\max} = \{v_i \in \mathcal{V}_c : L_i = \max\{L_1, \dots, L_n\}\}$ , and  $\mathcal{E}_c^{\max} = (\mathcal{V}_c^{\max} \times \mathcal{V}_c^{\max}) \cap \mathcal{E}$ .

**Lemma 6 (Maximal graph)** *Let  $\alpha^*$  be a min partition of the graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , and let  $\mathcal{G}^{\max} = (\mathcal{V}_c^{\max}, \mathcal{E}_c^{\max})$  be the maximal graph associated with  $\alpha^*$ . Then, for each  $v_i \in \mathcal{V}_c^{\max}, v_j \in \mathcal{V}_c \setminus \mathcal{V}_c^{\max}$  with  $\{v_i, v_j\} \in \mathcal{E}_c$ , it holds  $\alpha_{ij}^* = \underline{\alpha}_{ij}$  if  $i < j$ , and  $\alpha_{ij}^* = \bar{\alpha}_{ij}$  if  $i > j$ .*

**PROOF.** Let  $\alpha^*$  be a min partition, and let  $L^* = A\alpha^* + \mathbf{b}$ . By definition,  $\mathcal{V}_c^{\max}$  is the set of vertices  $v_i$  such that  $L_i^* = \|L^*\|_\infty$ . Let  $v_i \in \mathcal{V}_c^{\max}$ , and partition its neighbor set as  $\mathcal{N}_i^{\text{in}} = \mathcal{N}_i^1 \cup \mathcal{N}_i^2$ , where  $\mathcal{N}_i^2 = \mathcal{N}_i^{\text{in}} \cap \mathcal{V}_c^{\max}$  and  $\mathcal{N}_i^1 = \mathcal{N}_i^{\text{in}} \setminus \mathcal{N}_i^2$ . Suppose by contradiction that  $\alpha_{ij}^* > \underline{\alpha}_{ij}$  for some  $v_j \in \mathcal{N}_i^1$  with  $i < j$ . Define  $\hat{\alpha}$  from  $\alpha^*$  by modifying only the entry  $\hat{\alpha}_{ij} = \alpha_{ij}^* - \epsilon$ , with  $\epsilon \in \mathbb{R}_{>0}$ . Let  $\hat{L} = A\hat{\alpha} + \mathbf{b}$ , and let  $\epsilon$  be such that  $\hat{\alpha}_{ij} \geq \underline{\alpha}_{ij}$  and  $\hat{L}_i > \hat{L}_j$ . An equivalent condition for  $\hat{L}_i > \hat{L}_j$  is  $c_1 - c_2 > \ell_{ij}(\alpha_{ji}^* - \alpha_{ji}^* + 2\epsilon)$ , where  $c_1 = L_i^* - \alpha_{ij}^* \ell_{ij}$ ,  $c_2 = L_j^* - \alpha_{ji}^* \ell_{ij}$ , and  $\alpha_{ij} = 1 - \alpha_{ji}$  for  $i > j$ . It follows that

$$\begin{aligned} \|L^*\|_2^2 - \|\hat{L}\|_2^2 &= (L_i^*)^2 - \hat{L}_i^2 + (L_j^*)^2 - \hat{L}_j^2 = (c_1 + \alpha_{ij}^* \ell_{ij})^2 - (c_1 + \hat{\alpha}_{ij} \ell_{ij})^2 + (c_2 + \alpha_{ji}^* \ell_{ij})^2 - (c_2 + \hat{\alpha}_{ji} \ell_{ij})^2 \\ &= 2\epsilon \ell_{ij}(c_1 - c_2 + \ell_{ij}(\alpha_{ij}^* - \alpha_{ji}^* - \epsilon)) > 2\epsilon^2 \ell_{ij}^2 > 0, \end{aligned}$$

which contradicts our assumption of  $\alpha^*$  being a min partition. We conclude that  $\alpha_{ij} = \underline{\alpha}_{ij}$ . The case of  $\alpha_{ij}^* = \bar{\alpha}_{ij}$  is treated analogously, and the theorem follows.  $\square$

We are now ready to prove Theorem 5.

**PROOF.** Let  $\alpha^*$  be a min partition. Recall from Lemma 6 that there exists a set of cameras  $\mathcal{V}_c^{\max}$  such that (i)  $L_i^* = \|L^*\|_\infty$  for all  $v_i \in \mathcal{V}_c^{\max}$ , and (ii)  $\alpha_{ij}^* = \underline{\alpha}_{ij}$  (resp.  $\alpha_{ij}^* = \bar{\alpha}_{ij}$ ) if  $i < j$  (resp.  $i > j$ ) for all  $\{v_i, v_j\} \in \mathcal{E}_c$  with  $v_i \in \mathcal{V}_c^{\max}$  and  $v_j \in \mathcal{V}_c \setminus \mathcal{V}_c^{\max}$ . Notice also that, because of the cameras constraints and property (ii), it holds

$$\min_{\underline{\alpha} \leq \alpha \leq \bar{\alpha}} \sum_{v_i \in \mathcal{V}_c^{\max}} (A\alpha + \mathbf{b})_i \geq |\mathcal{V}_c^{\max}| \|L^*\|_\infty.$$

Let  $\hat{\alpha}$  be such that  $\|\hat{L}\|_\infty < \|L^*\|_\infty$ . Then  $\hat{L}_i < L_i^*$  for all  $v_i \in \mathcal{V}_c^{\max}$ . It follows that

$$\sum_{v_i \in \mathcal{V}_c^{\max}} (A\hat{\alpha} + \mathbf{b})_i < |\mathcal{V}_c^{\max}| \|L^*\|_\infty,$$

which contradicts our hypothesis.  $\square$

Distributed algorithms to compute continuous graphs partitions are presented in Section 6. In the next section we discuss the relationship between continuous graph partitions and the design of trajectories for camera network surveillance.

### 3 Setup for Camera Surveillance

In this section we describe our setup and we introduce some concepts which will be extensively used to state our results.

#### 3.1 Problem Setup

We consider the problem of surveilling an environment of interest by means of a camera network. We represent the environment with an undirected weighted roadmap  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V}$  and  $\mathcal{E}$  denote the vertex and the edge sets, respectively [15]. In particular, each vertex  $v_i \in \mathcal{V}$  corresponds to a location in the environment, and  $\{v_i, v_j\} \in \mathcal{E}$  if and only if the segment  $[v_i, v_j]$  joining vertices  $v_i$  and  $v_j$  belongs to the environment (vertices  $v_i$  and  $v_j$  are within line of sight). Finally, the weight of the edge  $\{v_i, v_j\} \in \mathcal{E}$  equals the length  $\ell_{ij} = \|v_i - v_j\|_2$ , and  $\ell^{\max} = \max\{\ell_{ij} : \{v_i, v_j\} \in \mathcal{E}\}$ .

Cameras are placed at the locations  $\mathcal{V}_c \subseteq \mathcal{V}$ . Define the set of *neighboring cameras*  $\mathcal{N}_i$  as in (1). The concept of neighboring cameras will be exploited in Section 6 to design distributed algorithms for the cameras.

Let  $x_i(t)$  denote the position at time  $t$  of the f.o.v. of the  $i$ -th camera. We assume that each camera has a limited visibility range along each adjacent edge. In particular,

- (A1) at all times  $t$ , the  $i$ -th f.o.v. is a point along the segment  $[v_i, v_j]$  for some  $v_j \in \mathcal{N}_i$ ;
- (A2) the speed of the  $i$ -th f.o.v. belongs to the set  $\{0, 1\}$ , that is, the f.o.v. of camera  $c_i$  either is stationary at some point or it moves at maximum (unitary) speed;
- (A3) for each  $v_j \in \mathcal{N}_i$ , a point  $v_{ij} \in [v_i, v_j]$  is given such that  $x_i(t) \in [v_i, v_{ij}]$  at all times  $t$ ;
- (A4) the cameras locations set  $\mathcal{V}_c$  satisfies

$$\bigcup_{v_i \in \mathcal{V}_c} \{\{v_i, v_j\} \in \mathcal{E} : v_j \in \mathcal{V}\} = \mathcal{E},$$

so that the roadmap  $\mathcal{G}$  is jointly visible by the cameras.

Our setup is illustrated in Fig. 1.

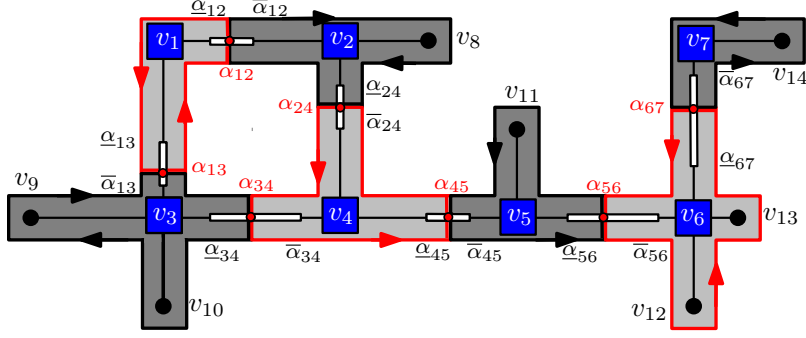


Fig. 1. This figure shows an environment to be surveilled by a camera network. The environment is represented by a roadmap  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with  $\mathcal{V} = \{v_1, \dots, v_{14}\}$ . Edges  $\mathcal{E}$  are denoted with solid black lines. Cameras are installed at the locations  $\mathcal{V}_c = \{v_1, \dots, v_7\}$ . White rectangles along the edges represent cameras visibility constraints, and the parameters  $\alpha$  define a continuous partition of  $\mathcal{G}$ . Finally, the DF-Trajectory associated with the partition given by  $\alpha$  is identified by the closed paths around the cameras.

### 3.2 Cameras trajectory

A *cameras trajectory* is a set of  $n$  continuous functions  $X = \{x_1, \dots, x_n\}$ , where  $x_i : \mathbb{R}_{\geq 0} \rightarrow \mathcal{E}$  describes the position of the  $i$ -th f.o.v. along the roadmap  $\mathcal{G}$ . We focus on *periodic* cameras trajectories, for which there exists a finite time  $T \in \mathbb{R}_{\geq 0}$  satisfying  $X(t+T) = X(t)$  for all  $t \in \mathbb{R}_{\geq 0}$ . Define the *image* of the  $i$ -th camera as the set of points visited by the  $i$ -th f.o.v. in any period of length  $T$ , i.e.,

$$\text{Im}(x_i) = \cup_{t \in [0, T]} x_i(t),$$

and the cameras *image set* as  $\mathcal{I}^X = \{\text{Im}(x_1), \dots, \text{Im}(x_n)\}$ .

---

#### Trajectory 1 DF-Trajectory for $i$ -th camera

---

**Input:** Parameters  $\alpha_{ij}$  and set of neighbors  $\mathcal{N}_i$ ;

Set  $s_i(t) = (tv_{ij} + (\alpha_{ij}\ell_{ij} - t)v_i)/\alpha_{ij}\ell_{ij}$ , for  $t \in [0, \alpha_{ij}\ell_{ij}]$ ;

Set  $t_0 = 0$ ;

**for**  $v_j \in \mathcal{N}_i$  **do**

Set  $v_{ij}$  as in Eq. (3);

$x_i(t) = s_i(t - t_0)$ , for  $t \in [t_0, t_0 + \alpha_{ij}\ell_{ij}]$ ;

$x_i(t) = s_i(2\alpha_{ij}\ell_{ij} - (t - t_0))$ , for  $t \in [t_0 + \alpha_{ij}\ell_{ij}, t_0 + 2\alpha_{ij}\ell_{ij}]$ ;

$t_0 = t_0 + 2\alpha_{ij}\ell_{ij}$ ;

**end for**

---

We now define a particular cameras trajectory associated with a continuous roadmap partition. The optimality properties of this trajectory will be shown in the subsequent sections. Let  $\alpha^{\text{df}}$  define the continuous partition  $\mathcal{P}^{\text{df}}$  as in (2). The *DF-Trajectory*  $X^{\text{df}}$  with image set  $\mathcal{P}^{\text{df}}$  is obtained by letting each camera sweep its subroadmap in a depth-first order [10], and it is formally described in Trajectory 1. See Fig. 1 for a graphical illustration.

### 3.3 Performance criteria

In this work we design cameras trajectories to detect intruders along the roadmap. We consider both *static*, and *dynamic* intruders. The trajectory of an intruder is a continuous function  $p : \mathbb{R}_{\geq 0} \rightarrow \mathcal{E}$ . Let  $\Pi_d$  be the set of all possible intruder trajectories, and let  $\Pi_s$  the set of static intruder trajectories ( $p(t) = p_0$  for all  $t \geq t_0$  and for some  $p_0 \in \mathcal{E}$ ).

An intruder is detected as soon as its position coincides with the f.o.v. of a camera. We define the *worst-case detection time* of a cameras trajectory as the longest time for the detection of an intruder. In particular, for an intruder appearing at time  $t_0$  and moving with trajectory  $p$ , and a cameras trajectory  $X$ , let

$$t^*(t_0, p, X) = \min \{ \{t - t_0 : t > t_0, p(t) \in X(t)\} \cup \{\infty\} \}.$$

We define the static worst-case detection time as

$$\text{WDT}_s(X) := \sup_{p \in \Pi_s, t_0 \in [0, T]} t^*(t_0, p, X), \quad (9)$$

and the dynamic worst-case detection time as

$$\text{WDT}_d(X) := \sup_{p \in \Pi_d, t_0 \in [0, T]} t^*(t_0, p, X). \quad (10)$$

For the ease of notation we define

$$\text{WDT}_s^* = \inf_{X \in \Omega} \text{WDT}_s(X), \quad \text{WDT}_d^* = \inf_{X \in \Omega} \text{WDT}_d(X).$$

We conclude this section by observing that  $\text{WDT}_s(X) \leq \text{WDT}_d(X)$  for any cameras trajectory  $X$ , and that for any periodic cameras trajectory  $X$ ,  $\text{WDT}_s(X) < \infty$  *if and only if* the entire roadmap is persistently surveilled by the cameras, i.e.,  $\mathcal{E} \subseteq \text{Im}(X)$ . Necessary and sufficient conditions for a trajectory to have finite dynamic detection time are discussed in Section 5.

## 4 Camera Trajectory for Static Intruders

This section contains our results for the detection of static intruders.

### 4.1 Main results for static intruders

Consider a roadmap  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with cameras locations  $\mathcal{V}_c$ . Let  $\mathcal{P}^* = \{\mathcal{P}_1^*, \dots, \mathcal{P}_n^*\}$  be a continuous partition of  $\mathcal{G}$  of cardinality  $|\mathcal{V}_c| = n$  with smallest dimension, that is

$$\max_{i \in \{1, \dots, n\}} L_i^* = \min_{\mathcal{P}} \max_{i \in \{1, \dots, n\}} L_i, \quad (11)$$

where  $\mathcal{P} = \{\mathcal{P}_1, \dots, \mathcal{P}_n\}$  is a continuous partition of  $\mathcal{G}$ . Let  $X^*$  be the DF-Trajectory associated with the partition  $\mathcal{P}^*$ . Recall that the roadmap  $\mathcal{G}$  with cameras locations  $\mathcal{V}_c$  is a *tree* if the induced graph  $\mathcal{G}(\mathcal{V}_c)$  contains no cycles, and it is a *ring* if  $\mathcal{G}(\mathcal{V}_c)$  consists of a single cycle [10].

**Theorem 7 (Static worst-case detection for DF-Trajectory)** *Consider a roadmap  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with cameras locations  $\mathcal{V}_c$  and  $|\mathcal{V}_c| = n$ . Let  $X^*$  be the DF-Trajectory associated with a continuous partition  $\mathcal{P}^*$  of  $\mathcal{G}$  with smallest dimension. Then,*

$$(i) \quad \text{WDT}_s(X^*) = 2\mathcal{L}(\mathcal{P}^*), \text{ and}$$

$$(ii) \quad \text{WDT}_s(X^*) \leq 2\text{WDT}_s^*.$$

Moreover, if  $\mathcal{G}$  is a tree or a ring, then  $\text{WDT}_s(X^*) = \text{WDT}_s^*$ .

In Theorem 7 we show that cameras trajectories designed from a continuous roadmap partition achieve detection performance within a constant factor of optimal. Since optimality is guaranteed for tree and ring roadmaps, we state the following conjecture.

**Conjecture 8 (Optimality for cyclic roadmaps)** *Motivated by our results in Theorem 7, we conjecture that  $\text{WDT}_s(X^*) = \text{WDT}_s^*$  also for cyclic roadmaps.*



#### 4.2 Proof of Theorem 7

In this section we derive a proof of Theorem 7. We start by introducing the necessary notation and some preliminary results. We define a relative order among the cameras as follows. Let  $\{c_1, \dots, c_n\}$  be the set of cameras. For the neighboring cameras  $c_i$  and  $c_j$  let the time  $t$  be such that the  $i$ -th f.o.v. and the  $j$ -th f.o.v. lie on the edge  $\{v_i, v_j\}$ . Then we say that  $c_i \leq c_j$  if  $\|x_i(t) - v_i\|_2 \leq \|x_j(t) - v_i\|_2$ . If  $x_i(t)$  and  $x_j(t)$  lie on different edges at time  $t$ , our convention is  $x_i(t) \leq x_j(t)$  for  $i < j$ . A camera trajectory is *order-invariant* if the relative order of the cameras is preserved over time.

**Theorem 9 (Order-invariant cameras trajectory)** *Given a roadmap  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with cameras locations  $\mathcal{V}_c$  and a cameras trajectory  $X$ , there exists an order-invariant cameras trajectory  $\bar{X}$  with  $\text{WDT}_s(X) = \text{WDT}_s(\bar{X})$ .*

**PROOF.** Let  $c_i$  and  $c_j$  be adjacent cameras, and assume the existence of  $t \geq 0$  such that  $x_i(t) = x_j(t)$ . Define  $t_{ij}^0 = \min\{t \geq 0 : x_i(t) = x_j(t)\}$  and, recursively,  $t_{ij}^n = \min\{t > t_{ij}^{n-1} : x_i(t) = x_j(t)\}, n \in \mathbb{N}$ . An order-invariant trajectory can be derived from  $X$  permuting the cameras labels as follows ( $k = 0, 1, \dots$ ):

$$\begin{aligned} \bar{x}_i(t) &= x_j(t) \text{ and } \bar{x}_j(t) = x_i(t) \text{ if } t_{ij}^{2k} \leq t \leq t_{ij}^{2k+1}, \\ \bar{x}_i(t) &= x_i(t) \text{ and } \bar{x}_j(t) = x_j(t) \text{ otherwise.} \end{aligned}$$

Each point along the roadmap is visited at the same times in  $X$  and  $\bar{X}$ , thus  $\text{WDT}_s(\bar{X}) = \text{WDT}_s(X)$ .  $\square$

In general, the images of neighboring cameras may overlap. A camera trajectory is called *non-overlapping* if for every pair  $c_i$  and  $c_j$ , it holds  $\text{Int}(\text{Im}(x_i)) \cap \text{Int}(\text{Im}(x_j)) = \emptyset$ , where  $\text{Int}$  denotes the interior of a set.

**Theorem 10 (Non-overlapping cameras trajectory for tree and ring roadmaps)** *Given a tree (resp. ring) roadmap  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with cameras locations  $\mathcal{V}_c$  and a cameras trajectory  $X$ , there exists an order-invariant and non-overlapping cameras trajectory  $\bar{X}$  with  $\text{WDT}_s(\bar{X}) \leq \text{WDT}_s(X)$ .*

**PROOF.** Without affecting generality, we assume that the trajectory  $X$  is order-invariant (cf. Theorem 9). We start by considering tree roadmaps, and we define the trajectory  $\bar{X}$  from  $X$  as follows.

Let  $a_1^0 = |\mathcal{N}_1^{\text{in}}| + 1$ ,  $a_i^0 = |\mathcal{N}_i^{\text{in}}|$  for  $i = 2, \dots, n$  and let  $\mathcal{P}_i^0 = \emptyset$  for every  $i = 1, \dots, n$ . Iteratively perform the following operations:

$$\begin{aligned} S^k &= \{v_i \in \mathcal{V}_c : a_i^{k-1} = 1\}, \\ \mathcal{P}_i^k &= \text{Cl}(\text{Im}(x_i) \setminus \left(\bigcup_{v_j \in \mathcal{N}_i^{\text{in}}} \mathcal{P}_j^{k-1}\right)), \text{ for each } v_i \in S^k, \\ a_j^k &= a_j^{k-1} - 1, \text{ for each } v_j \in \mathcal{N}_i^{\text{in}} \cup \{v_i\}, \end{aligned}$$

where  $\text{Cl}(\cdot)$  denotes the closure of a set, and  $k = 1, 2, \dots$ . After a finite number  $k_f$  of iterations, the set  $\mathcal{P}^{k_f} = \{\mathcal{P}_1^{k_f}, \dots, \mathcal{P}_n^{k_f}\}$  is a continuous partition of  $\mathcal{G}$ . Finally, define the trajectory  $\bar{X}$  as the DF-Trajectory on the partition  $\mathcal{P}^{k_f}$ . In the interest of space, we do not prove the convergence of the above procedure, and we provide instead an illustration of the final partition in Fig. 2.

We now show that  $\text{WDT}_s(\bar{X}) \leq \text{WDT}_s(X)$ . Consider camera  $c_i$ , let  $|\mathcal{N}_i^{\text{in}}| = n_i$ , and let  $\partial(\mathcal{P}_i^{k_f}) = \{v_{i1}, \dots, v_{in_i}\}$  be the boundary points of  $\mathcal{P}_i^{k_f}$ . By construction, there is only one boundary point, say  $v_{i2}$ , such that some points in the interior of  $[v_i, v_{i2}]$  may be visited by a camera adjacent to  $c_i$  with reference to trajectory  $X$  (cf. Fig. 2). Notice that every boundary point is visited by  $c_i$ . Without affecting generality, and possibly after relabeling the boundary points, let  $t_1$  be such that  $x_i(t_1) = v_{i1}$ , and  $t_2$  such that  $x_i(t_2) = v_{i2}$  and no other boundary point is visited in  $\text{Int}([t_1, t_2])$ . Notice that  $t_2 - t_1 \geq \|v_i - v_{i1}\|_2 + \|v_i - v_{i2}\|_2$ , and that every boundary point must be visited in the interval  $[t_2, t_1 + \text{WDT}_s(X)]$ . It follows  $\text{WDT}_s(X) \geq 2 \sum_{j=1}^{n_i} \|v_i - v_{ij}\|_2$ , and this must hold for each camera  $c_i$ . Finally notice that  $\text{WDT}_s(\bar{X}) = \max_{i \in \{1, \dots, n\}} 2 \sum_{j=1}^{n_i} \|v_i - v_{ij}\|_2$ , so that the statement follows. The case of a ring roadmap can be treated analogously.  $\square$

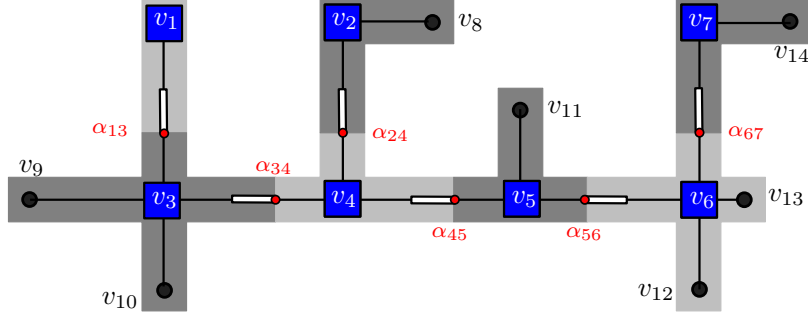


Fig. 2. This figure shows a tree roadmap, and it illustrates the partitioning procedure in the proof of Theorem 10.

We now prove Theorem 7.

**PROOF.** Statement (i) follows from the definition of DF-Trajectory, because each camera sweeps its assigned subroadmap at maximum speed along a depth-first tour.

Regarding statement (ii), consider a min partition  $\alpha$ , and let  $\mathcal{G}^{\max} = (\mathcal{V}_c^{\max}, \mathcal{E}_c^{\max})$  be its associated maximal graph (see Lemma 6). Define  $\text{Length}(\mathcal{G}^{\max}) = \sum_{\{v_i, v_j\} \in \mathcal{E}_c} \|v_i - v_j\|_2$ , and notice that

$$\text{WDT}_{s|\mathcal{G}^{\max}}^* \geq \frac{\text{Length}(\mathcal{G}^{\max})}{|\mathcal{V}_c^{\max}|},$$

where  $\text{WDT}_{s|\mathcal{G}^{\max}}^*$  denotes the smallest static worst-case detection time for  $\mathcal{G}^{\max}$ . Indeed, since  $L_i = L_j$  for each  $v_i, v_j \in \mathcal{V}^{\max}$ , each camera needs to sweep (at unitary speed) an image of length  $\frac{\text{Length}(\mathcal{G}^{\max})}{|\mathcal{V}_c^{\max}|}$  for  $\mathcal{G}^{\max}$  to be covered. Moreover, due to Lemma 6, cameras outside  $\mathcal{G}^{\max}$  cannot visit any point in the interior of  $\mathcal{G}^{\max}$ . It follows that

$$\text{WDT}_s^* \geq \text{WDT}_{s|\mathcal{G}^{\max}}^*.$$

Finally, since  $\text{WDT}_s(X^*) = 2 \frac{\text{Length}(\mathcal{G}^{\max})}{|\mathcal{V}_c^{\max}|}$ , we conclude that  $\text{WDT}_s(X^*) \leq 2\text{WDT}_s^*$ .

Consider a tree (resp. ring) roadmap. Due to Theorem 10, there exists an order-invariant and non-overlapping trajectory  $X$  with  $\text{WDT}_s(X) = \text{WDT}_s^*$ . To conclude the proof, we have  $\text{WDT}_s(X^*) \leq \text{WDT}_s(X)$  since  $\mathcal{I}^X$  is a continuous partition.  $\square$

## 5 Cameras trajectories for dynamic intruders

In this section we design cameras trajectories for the detection of dynamic intruders. We start by characterizing a necessary and sufficient condition on the cameras locations for the existence of trajectories with finite dynamic detection time.

**Theorem 11 (Existence of trajectories with finite dynamic detection time)** *Given a roadmap  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with cameras locations  $\mathcal{V}_c$ , the following statements are equivalent:*

- (i) *There exists a cameras trajectory  $X$  satisfying  $\text{WDT}_d(X) < \infty$ ;*
- (ii) *For every  $v_i \in \mathcal{V}_c$  with  $|\mathcal{N}_i| \geq 3$ , there exists  $v_j \in \mathcal{N}_i^{\text{in}}$  with  $\underline{\alpha}_{ij} = 0$  if  $i < j$  and  $\bar{\alpha}_{ij} = 1$  if  $i > j$ .*

The following result is useful to prove Theorem 11.

**Lemma 12 (Finite dynamic detection time for single camera)** *Given a roadmap  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with cameras locations  $\mathcal{V}_c = \{v_1\}$ , there exists a cameras trajectory  $X$  with  $\text{WDT}_d(X) < \infty$  if and only if  $|\mathcal{N}_1| \leq 2$ .*

**PROOF.** To show sufficiency, let  $|\mathcal{N}_1| \leq 2$ , and note that  $\mathcal{G}(\mathcal{V}_c)$  is a chain. Let  $x_1$  be such that camera  $c_1$  continuously sweeps the chain, and note that  $\text{WDT}_d(X) < \infty$ .

To show necessity of the statement, notice that if  $|\mathcal{N}_1| > 2$ , an intruder may choose its trajectory  $p$  so that  $p(t) \neq v_1$  and  $p(t + \varepsilon) \in [v_1, v_j]$  whenever  $x_1(t) = v_1$  and  $x_1(t + \varepsilon) \in [v_1, v_k]$ , for  $\varepsilon \in \mathbb{R}_{>0}$ ,  $v_j, v_k \in \mathcal{N}_1^{\text{out}}$ , and  $j \neq k$ .  $\square$

We are now ready to prove Theorem 11.

**PROOF.** In order to show that (ii) is a necessary condition for (i), suppose that (ii) does not hold. Then camera  $c_i$  needs to surveil a subroadmap in which  $|\mathcal{N}_i| \geq 3$ . The statement follows from Lemma 12.

We now show that (ii) is also a sufficient condition for (i) by proposing a procedure to clear every subroadmap from intruders appearing at time 0. By periodically repeating this procedure, intruders appearing at different times are also detected. Notice that intruders appearing along the edge  $[v_i, v_j]$  can be detected by moving the cameras  $c_i$  and  $c_j$  towards each other in a way that  $x_i(t) = x_j(t)$  for some time  $t$ . Starting from  $v_1$ , if  $|\mathcal{N}_1| \leq 2$ , then sweep the adjacent edges by synchronizing the motion of  $v_1$  and its neighboring cameras in any order. If  $|\mathcal{N}_1| > 2$ , suppose  $\{v_1, v_2\} \in \mathcal{V}_c$ , then let  $\underline{\alpha}_{12} = 0$  from condition (ii). Let camera  $c_2$  sweep the entire edge  $\{v_1, v_2\}$  and stop at  $v_1$ , and let the other neighboring cameras  $c_j \in \mathcal{N}_1^{\text{in}}$  stop at their vertices  $v_j$ . Then camera  $c_1$  sequentially sweeps its assigned adjacent segments  $[v_1, v_{1j}]$ , with  $\{v_1, v_j\} \in \mathcal{E}_c, j \neq 2$  by synchronizing its motion and those of the neighboring cameras, while keeping the f.o.v. of  $c_2$  at vertex  $v_1$ . Notice that any intruder appearing in an edge adjacent to  $v_1$  at time 0 is detected by this procedure. Then let  $c_1$  and its neighboring cameras return to their vertices. Finally iterate the procedure for subsequent cameras in increasing order, and repeat over time.  $\square$

In the remainder of this section we design cameras trajectories for the special cases of ring and tree roadmaps. We refer the reader to [21] for a solution to the case of chain roadmaps (cf. Equal-Waiting trajectory).

### 5.1 Ring roadmap

Consider a ring roadmap  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with cameras locations  $\mathcal{V}_c$ , and visibility constraints  $\underline{\alpha}, \bar{\alpha}$ . Let  $\mathcal{P}^* = \{\mathcal{P}_1^*, \dots, \mathcal{P}_n^*\}$  be a min-max partition of  $\mathcal{G}$ , and let  $\bar{L} = \mathcal{L}(\mathcal{P}^*) = \max\{L_1, \dots, L_n\}$ , where  $L_i$  is defined in (4). Notice that  $|\mathcal{N}_i^{\text{in}}| = 2$  for all  $v_i \in \mathcal{V}_c$ , and that each subroadmap  $\mathcal{P}_i^*$  can be written as a segment, parametrized by  $s_i : [0, L_i] \rightarrow [v_{i-1,i}, v_{i,i+1}]$ , such that  $s_i(t) = (tv_{i,i+1} + (L_i - t)v_{i-1,i})/L_i$ .

---

#### Trajectory 2 Ring-Sync-Trajectory for camera $c_i$

---

Camera  $c_i$  surveils  $\mathcal{P}_i^* = [v_{i-1,i}, v_i] \cup [v_i, v_{i,i+1}]$ ;  
Set  $I_n = [\bar{L}, 2\bar{L}]$ ,  $T_n = \bar{L}$ , if  $n$  is odd;  
 $I_n = \emptyset$ ,  $T_n = 0$ , if  $n$  even;  
**if**  $i$  is even **then**  
 $x_i(t) = v_{i-1,i}$ , for  $t \in [0, \bar{L} - L_i]$ ;  
 $x_i(t) = s_i(t - (\bar{L} - L_i))$ , for  $t \in [\bar{L} - L_i, \bar{L}]$ ;  
 $x_i(t) = v_{i,i+1}$ , for  $t \in I_n \cup [\bar{L} + T_n, 2\bar{L} - L_i + T_n]$ ;  
 $x_i(t) = s_i(2\bar{L} + T_n - t)$ , for  $t \in [2\bar{L} - L_i + T_n, 2\bar{L} + T_n]$ ;  
**else if**  $i$  is odd **then**  
 $x_i(t) = v_{i,i+1}$ , for  $t \in [0, \bar{L} - L_i]$ ;  
 $x_i(t) = s_i(\bar{L} - t)$ , for  $t \in [\bar{L} - L_i, \bar{L}]$ ;  
 $x_i(t) = v_{i-1,i}$ , for  $t \in I_n \cup [\bar{L} + T_n, 2\bar{L} - L_i + T_n]$ ;  
 $x_i(t) = s_i(t - (2\bar{L} - L_i + T_n))$ , for  $t \in [2\bar{L} - L_i + T_n, 2\bar{L} + T_n]$ ;  
**if**  $i = n$  **then**  
 $x_i(t) = v_{i,i+1}$ , for  $t \in [0, 2\bar{L} - L_i]$ ;  
 $x_i(t) = s_i(2\bar{L} - t)$ , for  $t \in [2\bar{L} - L_i, 2\bar{L}]$ ;  
 $x_i(t) = v_{i-1,i}$ , for  $t \in [2\bar{L}, 3\bar{L} - L_i]$ ;  
 $x_i(t) = s_i(t - (3\bar{L} - L_i))$ , for  $t \in [3\bar{L} - L_i, 3\bar{L}]$ ;  
**end if**  
**end if**

---

We propose the *Ring-Sync-Trajectory* in Algorithm 2; see Fig. 3 for a graphical illustration. We next provide an informal description of the Ring-Sync-Trajectory within its period for the case of an even number of cameras:

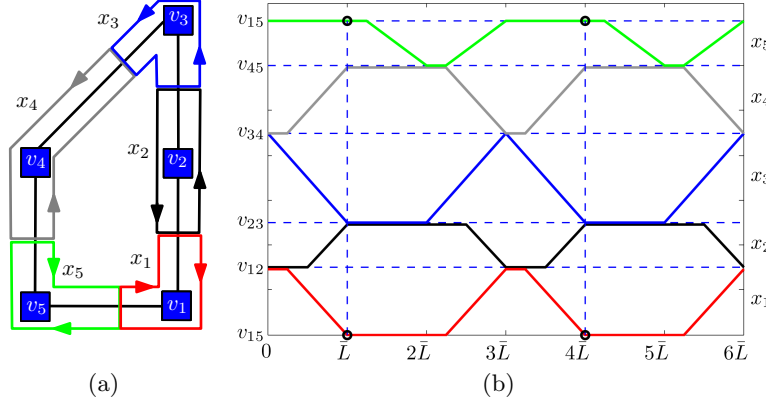


Fig. 3. Consider the ring roadmap  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with cameras locations  $\mathcal{V}_c$  and  $n = 5$ . Its associated optimal partition  $\mathcal{P}$  depicted in Fig. 3(a). The corresponding  $3\bar{L}$ - periodic Ring-Sync-Trajectory described in Trajectory 2 is shown in Fig. 3(b).

- (i) the f.o.v. of camera  $c_i$  is set at its right (left) boundary point if  $i$  is odd (even), so that all the cameras are pairwise synchronized;
- (ii) each camera sweeps its subroadmap at maximum speed, and
- (iii) each camera stops for  $\bar{L} - L_i$  at each boundary point.

If the number of cameras is odd, the algorithm is more involved and it is formally described in Trajectory 2. Notice that, (i) the  $i$ -th camera surveils only  $\mathcal{P}_i^*$ , (ii) the trajectory of each camera is  $3\bar{L}$ -periodic if  $n$  odd, and  $2\bar{L}$ -periodic if  $n$  even, and (iii) the f.o.v.s of each pair of neighboring cameras coincide at some times within the period.

**Theorem 13 (Dynamic worst-case detection for Ring-Sync-Trajectory)** *Given a ring roadmap  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with cameras locations  $\mathcal{V}_c$ , let  $X^s$  be the Ring-Sync-Trajectory in Algorithm 2. Then*

(i) if  $n$  is even,

$$\text{WDT}_d(X^s) = \text{WDT}_s(X^s) = \text{WDT}_d^*;$$

(ii) if  $n$  is odd,

$$\text{WDT}_d(X^s) = \text{WDT}_s(X^s) \leq \frac{3}{2} \text{WDT}_d^*.$$

**PROOF.** In order to prove statement (i), consider a ring roadmap with  $n$  even, and compute the optimal partition  $\mathcal{P}^*$  with dimension  $\bar{L}$  (see Section 6). Once we synchronize the cameras according to Trajectory 3, the minimum dynamic worst-case detection time is achieved. In order to prove (ii), notice that the Ring-Sync-Trajectory  $X^s$  is  $3\bar{L}$ -periodic when  $n$  is odd, and that each camera surveils the corresponding subroadmap  $\mathcal{P}_i^*$ . The thesis is proved observing that  $\text{WDT}_d^* \geq \text{WDT}_s^* = 2\bar{L}$ .  $\square$

## 5.2 Tree roadmap

Consider a tree roadmap  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with cameras locations  $\mathcal{V}_c = \mathcal{V} \setminus \{v_1\}$ , where node  $v_1$  is labeled as *root*. Assume that each camera can entirely surveil each adjacent edge, that is,  $\underline{\alpha} = \mathbf{0}$  and  $\bar{\alpha} = \mathbf{1}$ . We now design a cameras trajectory for dynamic intruders, and we show that the performance of our trajectory is within a constant factor of optimality. We start by recalling some definitions [10]. Then, vertex  $v_i \in \mathcal{V}_c$  is a *parent* of vertex  $v_j \in \mathcal{V}_c$  ( $v_i$  is a *child* of  $v_j$ ) if  $\{v_i, v_j\} \in \mathcal{E}_c$  and  $v_i$  lies on the unique shortest path from  $v_j$  to  $v_1$ . Let  $v_i^p$  denote the parent of  $v_i$ , and let  $\ell_i^p$  denote the length of the segment  $[v_i, v_i^p]$ . Recall that  $\ell^{\max}$  is the length of the longest edge. Let us parametrize the segment  $[v_i, v_i^p]$  as  $s_i : [0, \ell_i^p] \rightarrow [v_i, v_i^p]$ ,  $s_i(t) = (tv_i^p + (\ell_i^p - t)v_i)/\ell_i^p$ . We divide the vertices into two groups according to the parity of their distance to the root. In particular, we define  $\text{dist-r}(v_i)$  as the number of edges in the shortest path from  $v_i$  to the root  $v_1$ .

We propose the *Tree-Sync-Trajectory* in Trajectory 3 (see Fig. 4 for an example). An informal description of the Tree-Sync-Trajectory follows:

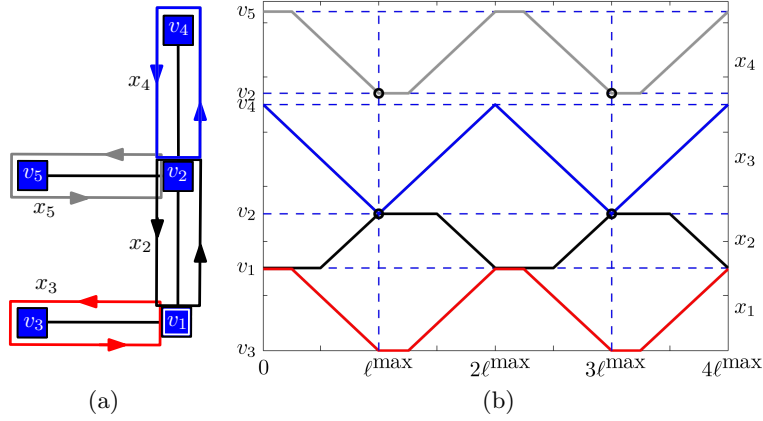


Fig. 4. Fig. 4(a) show a tree roadmap where  $v_1$  is labeled as root vertex. In Fig. 4(b) we report the Tree-Sync-Trajectory described in Trajectory 3. Note that cameras are synchronized, that is, for each pair of adjacent cameras  $\{v_i, v_j\} \in \mathcal{E}_c$ , there exists  $t \in [0, 2\ell^{\max}]$  such that  $x_i(t) = x_j(t)$ .

---

**Trajectory 3** Tree-Sync-Trajectory for camera  $c_i$

---

Camera  $c_i$  surveils only the segment  $[v_i, v_i^P]$ ;  
**if**  $\text{dist-r}(v_i)$  is odd **then**  
 $x_i(t) = v_i^P$ , for  $t \in [0, \ell_i^{\max} - \ell_i^P]$ ;  
 $x_i(t) = s_i(\ell_i^{\max} - t)$ , for  $t \in [\ell_i^{\max} - \ell_i^P, \ell_i^{\max}]$ ;  
 $x_i(t) = v_i$ , for  $t \in [\ell_i^{\max}, 2\ell_i^{\max} - \ell_i^P]$ ;  
 $x_i(t) = s_i(t - (2\ell_i^{\max} - \ell_i^P))$ , for  $t \in [2\ell_i^{\max} - \ell_i^P, 2\ell_i^{\max}]$ ;  
**else if**  $\text{dist-r}(v_i)$  is even **then**  
 $x_i(t) = v_i$ , for  $t \in [0, \ell_i^{\max} - \ell_i^P]$ ;  
 $x_i(t) = s_i(t - (\ell_i^{\max} - \ell_i^P))$ , for  $t \in [\ell_i^{\max} - \ell_i^P, \ell_i^{\max}]$ ;  
 $x_i(t) = v_i^P$ , for  $t \in [\ell_i^{\max}, 2\ell_i^{\max} - \ell_i^P]$ ;  
 $x_i(t) = s_i(2\ell_i^{\max} - t)$ , for  $t \in [2\ell_i^{\max} - \ell_i^P, 2\ell_i^{\max}]$ ;  
**end if**

---

- (i) the f.o.v. of camera  $c_i$  is set at the vertex  $v_i^P$  (resp.  $v_i$ ) if  $\text{dist-r}(v_i)$  is odd (resp. even),
- (ii) camera  $c_i$  sweeps the segment  $[v_i, v_i^P]$ , and
- (iii) camera  $c_i$  stops for  $\ell_i^{\max} - \ell_i^P$  at each boundary point.

Notice that, (i) camera  $c_i$  surveils only the segment  $[v_i, v_i^P]$ , (ii) the trajectory of each camera is  $2\ell^{\max}$ -periodic, (iii) cameras are *synchronized*, that is  $x_i(t_k) = x_i^P(t_k)$  for  $t_k = k\ell^{\max}$ , with  $k \in \mathbb{N}$  even (resp. odd) if  $\text{dist-r}(v_i)$  is even (resp. odd). Our Tree-Sync-Trajectory in Trajectory 3 extends the concept of Equal-waiting trajectory to the case of tree roadmap [21].

**Theorem 14 (Dynamic worst-case detection for Sync-Trajectory)** *Given a tree roadmap  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with cameras locations  $\mathcal{V}_c = \mathcal{V} \setminus \{v_1\}$  and  $\underline{\alpha} = \mathbf{0}$ ,  $\overline{\alpha} = \mathbf{1}$ , let  $X^s$  be the Tree-Sync-Trajectory in Trajectory 3. Then,*

$$\text{WDT}_d(X^s) = \text{WDT}_s(X^s) \leq 2\text{WDT}_d^*.$$

**PROOF.** Consider the Tree-Sync-Trajectory  $X^s$ . Notice that  $X^s$  is  $2\ell^{\max}$ -periodic, that each camera surveils a single edge, and that each edge is surveilled by a different camera. Let an intruder appear at time  $t_0$  along  $[v_i, v_i^P]$ , with  $\|v_i - p(t_0)\|_2 \leq \|v_i - x_i(t_0)\|_2$ . Then, such intruder is confined in  $[v_i, v_i^P] \cup (\cup_{v_j \text{ child of } v_i} [v_i, v_j])$  to avoid detection, where  $v_j$  is a child of  $v_i$ . Since  $v_i$  and all its children synchronize at most every  $2\ell^{\max}$ , we have  $\text{WDT}_d(X^s) = \text{WDT}_s(X^s) = 2\ell^{\max}$ . Notice that the case  $\|v_i - p(0)\|_2 > \|v_i - x_i(t_0)\|_2$  yields the same conclusion. On the other hand, since each edge can be surveilled by at most two cameras, it holds  $\text{WDT}_d^* \geq \text{WDT}_s^* \geq \ell^{\max}$ , which concludes the proof.  $\square$

## 6 Distributed Partitioning Algorithms

In what follows we design three distributed algorithms for the continuous min-max partitioning problem. Given an optimal partition, cameras organize along a DF-Trajectory as in Trajectory 1.

---

**Algorithm 4** Synchronous Gradient Partitioning

---

```
for  $v_i \in \mathcal{V}_c$  do
  Camera  $c_i$  receives  $S_j^t$  from  $c_j$ , for all  $v_j \in \mathcal{N}_i^{\text{in}}$ ;
  Set  $\alpha_{ij}^{t+1} \leftarrow \alpha_{ij}^t - \varepsilon \ell_{ij}(L_i^t - L_j^t)$ ;
  if  $\alpha_{ij}^{t+1} < \underline{\alpha}_{ij}$  then  $\alpha_{ij}^{t+1} = \underline{\alpha}_{ij}$ ;
  else if  $\alpha_{ij}^{t+1} > \bar{\alpha}_{ij}$  then  $\alpha_{ij}^{t+1} = \bar{\alpha}_{ij}$ ;
  end if
  Camera  $c_i$  transmits  $S_i^{t+1}$  to  $c_j$ , for all  $v_j \in \mathcal{N}_i^{\text{in}}$ .
end for
```

---

The algorithms we present rely upon different cameras communication assumptions. We assume each camera to be equipped with a wireless sensor device. In all our algorithms, cameras perform the following operations: (i) receive parameters from (some) neighboring cameras, (ii) update the parameters corresponding to (some) adjacent edges, and (iii) transmit the new values to (some) neighboring cameras. These operations are detailed in the next sections. For convenience, let  $S_i^t = \{\alpha_{ij}^t : v_j \in \mathcal{N}_i^{\text{in}}\}$  be the state of camera  $c_i$  at iteration  $t \in \mathbb{N}$ . Finally, we initialize  $\alpha_{ij}^0 = \underline{\alpha}_{ij}$  for all  $\{v_i, v_j\} \in \mathcal{E}_c$  with  $i < j$ .

### 6.1 Synchronous Gradient Partitioning algorithm

The distributed algorithm presented in this section assumes a synchronous mode of operation of the cameras, and it is inspired by the classical gradient projection method [5]. In particular, every camera performs operations at uniform time instants. The  $t$ -th iteration of this algorithm is detailed in Algorithm 4.

**Theorem 15 (Synchronous Gradient Partitioning)** *For a roadmap  $\mathcal{G}$  with cameras locations  $\mathcal{V}_c$ , let  $A$  and  $\mathbf{b}$  be as in (5) and (6), respectively. Let  $0 < \varepsilon < (d^{\max} \ell^{\max})^{-1}$ , where  $d^{\max} = \max\{|\mathcal{N}_i^{\text{in}}| : v_i \in \mathcal{V}_c\}$ . Then, the Synchronous Gradient Partitioning algorithm in Algorithm 4 asymptotically converges to  $\alpha_{\text{SGD}}^* = \lim_{t \rightarrow \infty} \alpha^t$ , and*

$$\min_{\underline{\alpha} \leq \alpha \leq \bar{\alpha}} \|A\alpha + \mathbf{b}\|_{\infty}^2 = \|A\alpha_{\text{SGD}}^* + \mathbf{b}\|_{\infty}^2,$$

where  $\underline{\alpha}$  and  $\bar{\alpha}$  denote the cameras constraints.

**PROOF.** Note that the update step can be expressed in vector form as

$$\alpha^{t+1} \leftarrow \alpha^t - \varepsilon A^{\top}(A\alpha^t + \mathbf{b}),$$

and that  $A^{\top}A\alpha + A^{\top}\mathbf{b}$  is the gradient of the quadratic function  $\frac{1}{2}\|A\alpha + \mathbf{b}\|_2^2$ . Therefore the Synchronous Partitioning algorithm coincides with the gradient projection method [5]. To conclude the proof note that the gradient of  $\frac{1}{2}\|A\alpha + \mathbf{b}\|_2^2$  is Lipschitz-continuous with some Lipschitz constant  $K \in \mathbb{R}_{>0}$ . Thus, for a sufficiently small step size  $\varepsilon$ , precisely if  $0 < \varepsilon < 2/K$ , the convergence of  $\alpha^t$  to a min partition is guaranteed by [5, Proposition 3.4]. In order to compute the stated upper bound for the stepsize  $\varepsilon$ , it can be shown that  $K \leq 2d^{\max}(\ell^{\max})^2$ . Finally, the claimed statement follows from Theorem 5. See [5] for further details.  $\square$

### 6.2 Asymmetric Broadcast Partitioning algorithm

The distributed algorithm presented in this section assumes an asymmetric broadcast communication protocol. In particular, at each iteration only one camera updates its state by using local information from its neighboring cameras. In order to guarantee the convergence of the algorithm, we assume the existence of a finite duration  $\tau \in \mathbb{R}_{>0}$  such that, for all  $t \in \mathbb{R}_{\geq 0}$ , every camera in  $\mathcal{V}_c$  is selected at least once in the time interval  $[t, t + \tau)$  (*partial asynchronism assumption*). The  $t$ -th iteration of this algorithm is detailed in Algorithm 5.

**Theorem 16 (Asymmetric Broadcast Partitioning)** *For a roadmap  $\mathcal{G}$  with cameras locations  $\mathcal{V}_c$  and cameras edges  $\mathcal{E}_c$ , let  $A$  and  $\mathbf{b}$  be as in (5) and (6), respectively. Let  $\tau$  be the partial asynchronism constant, and let  $0 <$*

---

**Algorithm 5** Asymmetric Broadcast Partitioning

---

Camera  $c_i$  is randomly selected;  
 Camera  $c_i$  receives  $S_j^t$  from camera  $c_j$ , for all  $v_j \in \mathcal{N}_i^{\text{in}}$ ;  
**for**  $v_j \in \mathcal{N}_i^{\text{in}}$  **do**  
 $\alpha_{ij}^{t+1} \leftarrow \alpha_{ij}^t - \varepsilon \ell_{ij} (L_i^t - L_j^t)$ ;  
**if**  $\alpha_{ij}^{t+1} < \underline{\alpha}_{ij}$  **then**  $\alpha_{ij}^{t+1} = \underline{\alpha}_{ij}$ ;  
**else if**  $\alpha_{ij}^{t+1} > \bar{\alpha}_{ij}$  **then**  $\alpha_{ij}^{t+1} = \bar{\alpha}_{ij}$ ;  
**end if**  
**end for**  
 Camera  $c_i$  transmits  $S_i^{t+1}$  to camera  $c_j$ , for all  $v_j \in \mathcal{N}_i^{\text{in}}$ .

---



---

**Algorithm 6** Symmetric Gossip Partitioning

---

Neighboring cameras  $c_i$  and  $c_j$  are randomly selected;  
 Camera  $c_i$  ( $c_j$ ) receives  $S_j^t$  ( $S_i^t$ ) from  $c_j$  ( $c_i$ );  
 $L^* = (L_i^t + L_j^t)/2$ ;  
 $\alpha_{ij}^{t+1} = (L^* - \sum_{v_k \in \mathcal{N}_{i,k}^{\text{in}}, k \neq j} \alpha_{ik}^t \ell_{ik}) / \ell_{ij}$ ;  
**if**  $\alpha_{ij}^{t+1} < \underline{\alpha}_{ij}$  **then**  $\alpha_{ij}^{t+1} = \underline{\alpha}_{ij}$ ;  
**else if**  $\alpha_{ij}^{t+1} > \bar{\alpha}_{ij}$  **then**  $\alpha_{ij}^{t+1} = \bar{\alpha}_{ij}$ ;  
**end if**  
 Camera  $c_i$  transmits  $S_i^{t+1}$  to  $c_k$ , for all  $v_k \in \mathcal{N}_i^{\text{in}}$ ;  
 Camera  $c_j$  transmits  $S_j^{t+1}$  to  $c_k$ , for all  $v_k \in \mathcal{N}_j^{\text{in}}$ .

---

$\varepsilon < (K(1 + \tau + \tau|\mathcal{E}_c|))^{-1}$ , where  $K \in \mathbb{R}_{>0}$  is the Lipschitz constant of  $\alpha \rightarrow A^T(A\alpha + \mathbf{b})$ . Then, the Asymmetric Broadcast Partitioning algorithm in Algorithm 5 asymptotically converges to  $\alpha_{\text{AB}}^* = \lim_{t \rightarrow \infty} \alpha^t$ . Moreover,

$$\min_{\underline{\alpha} \leq \alpha \leq \bar{\alpha}} \|A\alpha + \mathbf{b}\|_\infty^2 = \|A\alpha_{\text{AB}}^* + \mathbf{b}\|_\infty^2,$$

where  $\underline{\alpha}$  and  $\bar{\alpha}$  denote the cameras constraints.

**PROOF.** As in Theorem 15, the algorithm update follows the gradient of  $\alpha \rightarrow \frac{1}{2} \|A\alpha + \mathbf{b}\|_2^2$ . Because of the partial asynchronism assumption and the fact that  $\alpha^t \in [0, 1]^{\mathcal{E}_c}$  is such that  $\underline{\alpha} \leq \alpha^t \leq \bar{\alpha}$  for all  $t \in \mathbb{N}$ , the statement follows from [5, Section 7, Proposition 5.3] and Theorem 5. Note that the bound for the stepsize  $\varepsilon$  depends on the Lipschitz constant  $K$ , the time horizon  $\tau$  and the number of edges connecting cameras  $|\mathcal{E}_c|$ . See [5] for further details.  $\square$

### 6.3 Symmetric Gossip partitioning algorithm

The distributed algorithm presented in this section assumes a symmetric gossip-type communication protocol. In particular, at each time iteration only one component of a camera state is updated, and only two adjacent cameras are involved in the computation. The  $t$ -th iteration of this algorithm is detailed in Algorithm 6.

**Theorem 17 (Symmetric Gossip Partitioning)** For a roadmap  $\mathcal{G}$  with cameras locations  $\mathcal{V}_c$ , let  $A$  and  $\mathbf{b}$  be as in (5) and (6), respectively. Let the partial asynchronism assumption hold. Then, the Symmetric Gossip Partitioning algorithm in Algorithm 6 asymptotically converges to  $\alpha_{\text{SG}}^* = \lim_{t \rightarrow \infty} \alpha^t$ . Moreover,

$$\min_{\underline{\alpha} \leq \alpha \leq \bar{\alpha}} \|A\alpha + \mathbf{b}\|_\infty^2 = \|A\alpha_{\text{SG}}^* + \mathbf{b}\|_\infty^2,$$

where  $\underline{\alpha}$  and  $\bar{\alpha}$  denote the cameras constraints.

**PROOF.** Define  $U(\alpha) = \sum_{\{v_i, v_j\} \in \mathcal{E}_c} (L_i - L_j)^2$  as energy storage function. The convergence of the algorithm can be retrieved reasoning along the lines of [1, Theorem IV.1], and by applying Theorem 5.  $\square$

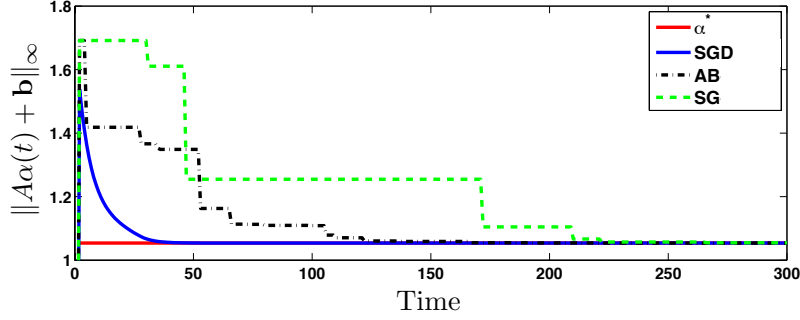


Fig. 5. This figure shows the convergence of the Synchronous Gradient Partitioning (SGD, blue solid line), the Asymmetric Broadcast Partitioning (AB, black dash-dot line), and the Symmetric Gossip Partitioning algorithms (SG, green dashed line) towards a solution of the continuous min-max partitioning problem. For the simulation we use the configuration in Fig. 2, with  $\underline{\alpha} = \mathbf{0}$ ,  $\bar{\alpha} = \mathbf{1}$  and  $\alpha^0 = \mathbf{0}$ .

To conclude this section, we validate our distributed algorithms through a numerical study. The tree roadmap considered for the simulations is in Fig. 2. Notice that  $|\mathcal{E}_c| = 6$ ,  $n = |\mathcal{V}_c| = 7$ , and the number of locations is given by  $|\mathcal{V}| = 14$ . The stepsizes for Algorithm 5 and Algorithm 6 are chosen sharp, up to a constant  $\epsilon = .01$ , to their upper bounds stated in Theorem 15 and Theorem 16, respectively. The results of our simulation study are in Fig. 5. Notice that all the proposed algorithms converge to the desired value.

## 7 Conclusion

In this work we design surveillance trajectories for a network of autonomous cameras. We consider both static and dynamic intruders in the environment to be surveilled. As performance criteria we consider the worst-case detection time of intruders. For the case of static intruders, we derive optimal trajectories for ring and tree roadmaps, and constant-factor suboptimal trajectories for general roadmaps. For the case of dynamics intruders, we derive constant-factor suboptimal trajectories for chain, ring, and tree roadmaps. As a complementary result, we introduce the continuous partitioning problem of a weighted graph, and we propose distributed algorithms for its solution.

## References

- [1] R. Alberton, R. Carli, A. Cenedese, and L. Schenato. Multi-agent perimeter patrolling subject to mobility constraints. In *American Control Conference*, pages 4498–4503, Montréal, Canada, June 2012.
- [2] K. Andreev and H. Racke. Balanced graph partitioning. *Theory of Computing Systems*, 39:929–939, 2006.
- [3] E. M. Arkin, R. Hassin, and A. Levin. Approximations for minimum and min-max vehicle routing problems. *Journal of Algorithms*, 59(1):1–18, 2006.
- [4] M. Baseggio, A. Cenedese, P. Merlo, M. Pozzi, and L. Schenato. Distributed perimeter patrolling and tracking for camera networks. In *IEEE Conf. on Decision and Control*, pages 2093–2098, Atlanta, GA, USA, December 2010.
- [5] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.
- [6] D. Borra, F. Pasqualetti, and F. Bullo. Continuous graph partitioning for camera network surveillance. In *IFAC Workshop on Distributed Estimation and Control in Networked Systems*, Santa Barbara, CA, USA, September 2012. To appear.
- [7] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [8] R. Carli, A. Cenedese, and L. Schenato. Distributed partitioning strategies for perimeter patrolling. In *American Control Conference*, pages 4026–4031, San Francisco, CA, USA, June 2011.
- [9] G. Cybenko. Dynamic load balancing for distributed memory multiprocessors. *Journal of Parallel and Distributed Computing*, 7(2):279–301, 1989.
- [10] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, 2 edition, 2000.
- [11] G. Even, N. Garg, J. Könemann, R. Ravi, and A. Sinha. Min-max tree covers of graphs. *Operations Research Letters*, 32(4):309–315, 2004.
- [12] G. Even, J. Naor, S. Rao, and B. Schieber. Fast approximate graph partitioning algorithms. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 639–648, New Orleans, LA, USA, January 1997.



- [13] A. Kehagias, G. Hollinger, and S. Singh. A graph search algorithm for indoor pursuit/evasion. *Mathematical and Computer Modelling*, 50(9-10):1305–1317, 2009.
- [14] A. Kolling and S. Carpin. Pursuit-evasion on trees by robot teams. *IEEE Transactions on Robotics*, 26(1):32–47, 2010.
- [15] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006. Available at <http://planning.cs.uiuc.edu>.
- [16] R. Lüling, B. Monien, and F. Ramme. Load balancing in large networks: A comparative study. In *IEEE Symposium on Parallel and Distributed Processing*, pages 686–689, Dallas, TX, USA, December 1991.
- [17] A. Machado, G. Ramalho, J. D. Zucker, and A. Drogoul. Multi-agent patrolling: An empirical analysis of alternative architectures. In *Multi-Agent-Based Simulation II*, Lecture Notes in Computer Science, pages 155–170. Springer, 2003.
- [18] J. O’Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.
- [19] T. D. Parsons. Pursuit-evasion in a graph. In Y. Alavi and D. Lick, editors, *Theory and Applications of Graphs*, volume 642 of *Lecture Notes in Mathematics*, pages 426–441. Springer, 1978.
- [20] F. Pasqualetti, A. Franchi, and F. Bullo. On cooperative patrolling: Optimal trajectories, complexity analysis and approximation algorithms. *IEEE Transactions on Robotics*, 28(3):592–606, 2012.
- [21] M. Spindler, F. Pasqualetti, and F. Bullo. Distributed multi-camera synchronization for smart-intruder detection. In *American Control Conference*, pages 5120–5125, Montréal, Canada, June 2012.
- [22] S. Susca, S. Martínez, and F. Bullo. Monitoring environmental boundaries with a robotic sensor network. *IEEE Transactions on Control Systems Technology*, 16(2):288–296, 2008.